Opérateur DYNA_TRAN_MODAL

1 But

Calculer la réponse dynamique transitoire d'un système amorti ou non en coordonnées généralisées. Le calcul est effectué par superposition modale ou par sous-structuration.

Des conditions initiales non-nulles peuvent être introduites permettant entre autre d'utiliser les résultats d'un calcul antérieur.

Le chargement est donné sous la forme d'une combinaison linéaire de vecteurs généralisés et de fonctions du temps décrivant l'évolution temporelle de ces vecteurs.

Six méthodes d'intégration explicites : 'EULER', 'DEVOGE', 'ADAPT_ORDRE1' et 'ADAPT_ORDRE2' (méthodes d'intégration à pas de temps adaptatif d'ordre 1 ou d'ordre2), 'RUNGE_KUTTA_32' et 'RUNGE_KUTTA_54' (méthodes d'intégration à pas de temps adaptatif de la famille de Runge-Kutta à ordres 54 et 32 respectivement), une méthode intégrale 'ITMI' et une méthode d'intégration implicite : 'NEWMARK' sont disponibles. Les algorithmes explicites et 'ITMI' supportent le calcul avec prise en compte de non-linéarités localisées aux nœuds de type chocs et frottement. Les méthodes 'EULER', 'ADAPT_ORDRE1' et 'ADAPT_ORDRE2' supportent la prise en compte de non-linéarités de type dispositif anti-sismique. Les méthodes 'RUNGE_KUTTA_32', et 'RUNGE_KUTTA_54' supportent la prise en compte de non-linéarités localisées aux nœuds de type chocs et frottement ainsi que des non-linéarités de type dispositif anti-sismique.

Pour pouvoir calculer la réponse d'une machine tournante sur paliers non-linéaires, un coupleur avec le code EDYOS a été développé et nécessite l'utilisation conjointe des mot-clés facteurs COUPLAGE_EDYOS et PALIER_EDYOS.

La structure de données resultat contient pour différents instants de calcul, les résultats généralisés et les forces de choc calculées.

La conversion des résultats généralisés dans l'espace physique est possible par les opérateurs REST_GENE_PHYS [U4.63.31] ou pour une composante par RECU_FONCTION [U4.32.03].

Produit un concept de type tran_gene.

Table des matières

2 Syntaxe

```
tranmo [tran_gene] = DYNA_TRAN_MODAL    (

        ◊  reuse =   tranmo,

         ♦ MATR_MASS =        ma  ,                       [matr_asse_gene_R]
         ♦ MATR_RIGI =        ri  ,                       [matr_asse_gene_R]
         ◊ MATR_AMOR =        am  ,                       [matr_asse_gene_R]
         ◊ AMOR_MODAL = _F(
                           /  AMOR_REDUIT =    la  ,      [l_R]
                           /  LIST_AMOR   =    l_amor ,   [listr8]
                        ),
         ◊ SCHEMA_TEMPS =_F(
                           ♦ SCHEMA = ( |'NEWMARK',      [DEFAUT]
                                        |'EULER',
                                        | 'DEVOGE',
                                        | 'ADAPT_ORDRE1',
                                        | 'ADAPT_ORDRE2',
                                        | 'RUNGE_KUTTA_54',
                                         | 'RUNGE_KUTTA_32',
                                        | 'ITMI',

                              ),
```

\# Mots clés associés uniquement au schéma 'NEWMARK':

```
                           ◊ BETA    =/0.25,             [DEFAUT]
                                      /beta,             [R]
                           ◊ GAMMA   =/0.5,              [DEFAUT]
                                      /gamma,            [R]
```

\# Mots clés associés uniquement aux schémas 'RUNGE_KUTTA_*':

```
                           ◊ TOLERANCE  =/1.E-3,         [DEFAUT]
                                         /tol,           [R]
                           ◊ ALPHA      =/1.E-3,         [DEFAUT]
                                         /alpha,         [R]
```

\# Mots clés associés uniquement au schéma 'ITMI' :

```
                           ◊  BASE_ELAS_FLUI=   meles,   [melasflu]
                           ◊  NUME_VITE_FLUI=   Nvitf,   [I]
                           ◊  ETAT_STAT     =  /'NON',   [DEFAUT]
                                               /'OUI',
                           ◊  PREC_DUREE    =  /1.E-2,   [DEFAUT]
                                               /prec,    [R]
                           ◊  CHOC_FLUI     =  /'NON',   [DEFAUT]
                                 /  'OUI',
                           ◊  NB_MODE       =  Nmode,    [I]
                           ◊  NB_MODE_FLUI  =  Nmodef,   [I]
                           ◊  TS_REG_ETAB   =  tsimu,    [R]
                              ),

        ♦   INCREMENT =_F(  ♦   /  LIST_INST = litps,     [listr8]
                                /  PAS       = dt,        [R]
                           ◊  INST_INIT  = ti,            [R]
                           ◊  /  INST_FIN= tf,            [R]
                              /  NUME_FIN= nufin,         [I]

                           ◊  VERI_PAS =    / 'OUI',      [DEFAUT]
                                            / 'NON',
```

\# Opérandes spécifiques à une intégration par pas de temps adaptatifs d'ordre 1 et 2

```
                           ◊  VITE_MIN =       / 'NORM',     [DEFAUT]
                                               / 'MAXI',
```

```
                              ◊  COEF_MULT_PAS =  / 1.1  ,       [DEFAUT]
                                                  / cmp  ,       [R]
                              ◊  COEF_DIVI_PAS =  / 1.33333334,  [DEFAUT]
                                                  /  cdp ,       [R]
                              ◊  PAS_LIMI_RELA =  /  1.E-6,      [DEFAUT]
                                                  /  per ,       [R]
                              ◊  NB_POIN_PERIODE =/  50,         [DEFAUT]
                                                  /  N,          [I]
                              ◊  NMAX_ITER_PAS =  /  16,         [DEFAUT]
                                                  /  N,          [I]
```

Opérandes spécifiques à une intégration par pas de temps adaptatifs (ordre 1 et 2) ainsi que Runge-Kutta

```
                              ◊  PAS_MAXI =          dtmax,      [R]
                              ◊  PAS_MINI =          dtmin,      [R]  ),
      ◊ ETAT_INIT = _F(   ♦  /  RESULTAT =res,              [tran_gene]
                              .. Si RESULTAT
                                   ◊  /INST_INIT =  to,        [R]
                                      /NUME_ORDR =  no,        [I]
                                   ◊  / CRITERE =   'RELATIF', [DEFAUT]
                                   ◊ PRECISION = / 1.E-06,     [DEFAUT]
                                                 / prec,       [R]
                                      / CRITERE =    'ABSOLU',
                                   ♦ PRECISION = prec,         [R]

                                 /    | DEPL = do,              [vect_asse_gene]
                                      | VITE = vo,              [vect_asse_gene]
                              ),
      ◊ EXCIT =   _F( ◊  VECT_ASSE_GENE = v,            [vect_asse_gene]
                      ◊  NUME_ORDRE = nmordr,           [I]
                      ◊      /  FONC_MULT =  f,         [fonction]
                             /   COEF_MULT = a,         [R]
                             /  ◊  ACCE = ac,           [fonction]
                                ◊  VITE = vi,           [fonction]
                                ◊  DEPL = dp,           [fonction]
```

Opérandes et mots clés spécifiques à l'analyse sismique

```
                      ◊  MULT_APPUI = /  'NON',    [  DEFAUT]
                                      /  'OUI',
                      ◊  DIRECTION =(dx,dy,dz,drx,dry,drz),   [l_R]
                      ◊  /   NOEUD =      lno,                [l_noeud]
                         /   GROUP_NO  =  lgrno,              [l_groupe_no]

                      ◊  ♦ CORR_STAT  =   'OUI'
                         ♦ D_FONC_DT  =   dfdt,               [fonction]
                         ♦ D_FONC_DT2 =   dfdt2,              [fonction]

                  ),

      ◊      / MODE_STAT = psi,                           [mode_meca]
             / MODE_CORR = modcor,              [mult_elas, mode_meca ]

      ◊ EXCIT_RESU    =
        _F(    ♦   RESULTAT      =  resuforc,             [tran_gene]
               ◊   /COEF_MULT    =  ai,                   [R]
                   /COEF_MULT_C  =  aci,                  [C]
             ),
```

Fin des opérandes et mots clés spécifiques à l'analyse sismique

```
      ◊  CHOC = _F(
                   ◊  INTITULE =   int,                   [l_Kn]

                   /  ♦  /  NOEUD_1      =  no1,          [noeud]
```

```
                        /   GROUP_NO_1    =  grno1,           [group_no]
                    ◊   /   NOEUD_2       =  no2,             [noeud]
                        /   GROUP_NO_2    =  grno2,           [group_no]
                /   ♦   /   MAILLE        =  ma,              [maille]
                        /   GROUP_MA      =  grma,            [group_ma]

                ♦   OBSTACLE =    obs,                        [obstacle]
                ♦   NORM_OBST =   nor,                        [listr8]
                ◊   ORIG_OBST =   ori,                        [listr8]
                ◊   JEU =         / 1.,                       [DEFAUT]
                                  / jeu,                      [R]

                ◊   ANGL_VRIL =   gamma,                      [R]

                ◊   DIST_1 =  dist1,                          [R]
                ◊   DIST_2 =  dist2,                          [R]

                ◊   SOUS_STRUC_1 =   ss1,                     [K8]
                ◊   SOUS_STRUC_2 =   ss2 ,                    [K8]
                ◊   REPERE =      / 'GLOBAL',                 [DEFAUT]
                                  / nom_sst,                  [K8]

                ◊   RIGI_NOR =    kn,                         [R]
                ◊   AMOR_NOR =    / 0.,                       [DEFAUT]
                                  / cn,                       [R]
                ◊   RIGI_TAN =    / 0.,                       [DEFAUT]
                                  / kt,                       [R]
                ◊   AMOR_TAN =    / ct,                       [R]
                ◊   FROTTEMENT =
                                  / 'NON'                     [DEFAUT]
                                  / 'COULOMB'
                                     ♦ COULOMB = mu       [R]
                                  / 'COULOMB_STAT_DYNA'
                                     ♦ COULOMB_STAT = mus     [R]
                                     ♦ COULOMB_DYNA = mud     [R]


    # Opérandes spécifiques à la prise en compte d'un transitoire de vitesse
      pour les rotors (vitesse de rotation variable)
        ◊ VITESSE_VARIABLE =/ 'NON',                          [DEFAUT]
                            / 'OUI',
          # si VITESSE_VARIABLE='OUI' :
                ♦   VITE_ROTA =  vrota,                   [fonction]
                ♦   MATR_GYRO =  gyro,                    [matr_asse_gene_R]
                ◊   ACCE_ROTA =  arota,                   [fonction]
                ◊   MATR_RIGY =  gyro,                    [matr_asse_gene_R]
          # si VITESSE_VARIABLE='NON' :
                ♦   VITE_ROTA =  / 0.0,                   [DEFAUT]
                                 / vrota,                 [R]


    # Mot-clef spécifique à la prise en compte d'une fissure dans un rotor
       ◊  ROTOR_FISS=_F(
                /   ♦   /   NOEUD_G       =  nog,             [noeud]
                        /   GROUP_NO_G    =  grnog,           [group_no]
                    ♦   /   NOEUD_D       =  nod,             [noeud]
                        /   GROUP_NO_D    =  grnod,           [group_no]
                ♦   ANGL_INIT =   0.0,                        [DEFAUT]
                ◊   ANGL_ROTA =   0.0,                        [fonction]
                ♦   K_PHI     =   kphi                        [fonction]
                ♦   DK_DPHI   =   dkdphi                      [fonction]
                 )


          ◊   VERI_CHOC =    _F(
                ◊   STOP_CRITERE  =  /   'OUI',               [DEFAUT]
                                     /   'NON',
                ◊   SEUIL =   /  0.5 ,                        [DEFAUT]
                             /  s,                            [R]
```

```
                        ),

          ◊  ANTI_SISM =    _F(
                       ♦ /  NOEUD_1      =  no1,                  [noeud]
                         /  GROUP_NO_1   =  grno1,                [group_no]
                       ♦ /  NOEUD_2      =  no2,                  [noeud]
                         /  GROUP_NO_2   =  grno2,                [group_no]
                       ◊ RIGI_K1 =     / 0.,                      [DEFAUT]
                                       / kn,                      [R]
                       ◊ RIGI_K2 =     / 0.,                      [DEFAUT]
                                       / kn,                      [R]
                       ◊ SEUIL_FX =    / 0.,                      [DEFAUT]
                                       / Py,                      [R]
                       ◊ C =           / 0.,                      [DEFAUT]
                                       / C,                       [R]
                       ◊ PUIS_ALPHA =  / 0.,                      [DEFAUT]
                                       / alpha,                   [R]
                       ◊ DX_MAX =      / 1.,                      [DEFAUT]
                                       / dx,                      [R]
                         ),
      ◊    DIS_VISC = _F(
                ♦   / NOEUD_1          = no1,                        [noeud]
                    / GROUP_NO_1       = grno1,                   [group_no]
                ♦   / NOEUD_2          = no2,                        [noeud]
                    / GROUP_NO_2       = grno2,                   [group_no]

                ♦   / K1               = k1,                             [R]
                    / UNSUR_K1         = usk1,                           [R]
                ♦   / K2               = k2,                             [R]
                    / UNSUR_K2         = usk2,                           [R]
                ♦   / K3               = k3,                             [R]
                    / UNSUR_K3         = usk3,                           [R]
                ♦   C                  = c,                              [R]
                ♦   PUIS_ALPHA         = / 0.5                      [defaut]
                                         / alpha,                        [R]

                ◊   ITER_INTE_MAXI     = / 20                       [defaut]
                                         / iter                          [I]
                ◊   RESI_INTE_RELA     = / 1.0E-06                  [defaut]
                                         / resi                          [R]
           ),

           ◊ FLAMBAGE =    _F(
                   ♦  /  NOEUD_1      =  no1,                     [noeud]
                      /  GROUP_NO_1   =  grno1,                   [group_no]
                   ◊  /  NOEUD_2      =  no2,                     [noeud]
                      /  GROUP_NO_2   =  grno2,                   [group_no]
                   ♦  OBSTACLE = obs,                             [obstacle]
                   ◊  ORIG_OBST = ori,                            [listr8]
                   ♦  NORM_OBST = nor,                            [listr8]
                   ◊  ANGL_VRIL =  / 0,                           [DEFAUT]
                                   / gamma,                       [R]
                   ◊  JEU =  / 1.,                                [DEFAUT]
                             /jeu,                                [R]
                   ◊  DIST_1 = dist1,                             [R]
                   ◊  DIST_2 = dist2,                             [R]
                   ◊  REPERE =  /'GLOBAL',                        [DEFAUT]
                                / nom_sst ,                       [K8]
                   ◊  RIGI_NOR = kn,                              [R]
                   ◊  FNOR_CRIT = flim,                           [R]
                   ◊  FNOR_POST_FL = fseuil,                      [R]
                   ◊  RIGI_NOR_POST_FL = k2,                      [R]
                   ),

          ◊  RELA_EFFO_DEPL = _F(
                       ♦   NOEUD =     noe,                       [noeud]
```

```
                ◊   SOUS_STRUC = ss,                [K8]
                ◊   NOM_CMP =    nomcmp,            [K8]
                ♦   RELATION =   f,                 [fonction]
                       ),

    ◊   RELA_EFFO_VITE = _F(
                ♦    NOEUD =     noe,               [noeud]
                ◊   SOUS_STRUC = ss,                [K8]
                ◊   NOM_CMP=     nomcmp,            [K8]
                ♦   RELATION =   f,                 [fonction]
                       ),
```

Mots clés facteurs associés uniquement au couplage avec le code EDYOS

```
    ◊   COUPLAGE_EDYOS   =   _F(
                ♦   VITE_ROTA        =      vrota,        [R]
                ♦   PAS_TPS_EDYOS    =   dtedyos,         [R]
                       ),

    ◊   PALIER_EDYOS =   _F(
                ♦   /  UNITE     =   uled,               [I]
                    /  GROUP_NO  =   grnoed,             [group_no]
                     / NOEUD     =   noed,               [noeud]
                ◊ TYPE_EDYOS     =   /  'PAPANL',
                                     /  'PAFINL',
                                     /  'PACONL',
                                     /  'PAHYNL',
                       ),
```

Fin des mots clés facteurs associé uniquement au couplage avec le code EDYOS

```
      ◊ ARCHIVAGE =_F(   ◊  / LIST_INST  = list      [listr8]
                            / INST       = in        [R]
                            / PAS_ARCH   = ipa       [I]
                         ◊ / CRITERE =   'RELATIF',  [DEFAUT]
                           ◊ PRECISION = / 1.E-06, [  DEFAUT]
                                         / prec,     [R]
                           / CRITERE =   'ABSOLU',
                           ♦ PRECISION = prec,       [R]
                     ),

     ◊ SOLVEUR = _F (voir [U4.50.01])

    ◊   INFO      =  /  1,                           [DEFAUT]
                     /  2,

    ◊   IMPRESSION = _F(
              ◊  /  TOUT = 'OUI',                    [DEFAUT]
                 /  NIVEAU =  |  'DEPL_LOC',
                              |  'VITE_LOC',
                              |  'FORC_LOC',
                              |  'TAUX_CHOC',
              ◊  INST_INIT = ti,                     [R]
              ◊  INST_FIN =  tf,                     [R]
              ◊   UNITE_DIS_VISC =                   [I]
                    ),

    ◊   TITRE     =  titre,                          [l_Kn]
)
```

3 Opérandes

3.1 Matrices généralisées

Dans le cas d'un calcul par recombinaison modale, les matrices généralisées doivent être établies par l'opérateur PROJ_MATR_BASE [U4.63.12] ou par la macro-commande PROJ_BASE [U4.63.11], à partir de la même base modale.
Dans le cas d'un calcul par sous-structuration dynamique, les matrices généralisées doivent être établies par l'opérateur ASSE_MATR_GENE [U4.65.04], à partir de la même numérotation généralisée.

♦ MATR_MASS = ma

Matrice de masse du système généralisé.
Concept de type matr_asse_gene_R.

♦ MATR_RIGI = ri

Matrice de rigidité du système généralisé.
Concept de type matr_asse_gene_R.

◊ MATR_AMOR = am

Matrice d'amortissement du système généralisé.
Concept de type matr_asse_gene_R.
Cette option n'est pas disponible avec la méthode 'DEVOGE'.

3.2 Mot-clé AMOR_MODAL

Ce mot-clé permet de prendre en compte un amortissement équivalent à de l'amortissement modal décomposé sur une base de modes pré-calculée sous forme de concept de type mode_meca. Cet amortissement est globalement pris en compte dans l'équation d'équilibre dynamique comme une force correctrice au second membre $-\mathbf{C}\dot{\mathbf{X}}$.

3.2.1 Opérandes AMOR_REDUIT / LIST_AMOR

◊ / AMOR_REDUIT = l η

Liste des amortissements réduits ($\eta_{1,}\eta_{2,}\ldots,\eta_n$ pourcentages de l'amortissement critique) correspondants à chaque mode du système sous forme de liste de réels.
Cette option n'est pas disponible en sous-structuration dynamique car les amortissements réduits doivent être définis pour chaque sous-structure séparément (opérateur MACR_ELEM_DYNA [U4.65.01]).

Remarque :

Si le nombre d'amortissements réduits donnés est inférieur au nombre de vecteurs de base utilisés dans la base modale, les amortissements des vecteurs supplémentaires sont pris égaux au dernier amortissement de la liste.

/ LIST_AMOR = c_formule η

Nom du concept de type listr8 contenant la liste des amortissements réduits.

3.3 Schémas d'intégration. Mot clé SCHEMA_TEMPS

Sous ce mot-clé on peut renseigner un schéma d'intégration avec, éventuellement, ses paramètres. Les schémas disponible sont à déclarer sous l'opérande SCHEMA.

3.3.1 Opérande SCHEMA

◊ SCHEMA =

Choix de la méthode numérique de résolution.
Dans le cas d'un calcul classique par recombinaison modale, l'utilisateur dispose de six méthodes de type explicite, d'une méthode intégrale et d'une méthode de type implicite.
Dans le cas d'un calcul par sous-structuration dynamique [R4.06.04], la méthode de calcul transitoire sur base modale calculée par sous-structuration supporte tous les schémas d'intégration évoqués sauf la méthode intégrale. En revanche, la méthode de calcul transitoire sur les "bases" des sous-structures ne supporte que le schéma d'Euler et les schémas à pas de temps adaptatif.

3.3.1.1 SCHEMA = 'NEWMARK' : schéma implicite

Ce schéma ne permet que l'intégration de problèmes linéaires. C'est le schéma par défaut pour la résolution. On peut préciser les paramètres d'intégration β et γ :

◊ BETA = beta

Valeur du paramètre β pour la méthode de NEWMARK. Par défaut $\beta=0.25$.

◊ GAMMA = gamm

Valeur du paramètre γ pour la méthode de NEWMARK. Par défaut $\gamma=0.5$.

3.3.1.2 SCHEMA = 'EULER' : schéma explicite d'ordre 1

Ce schéma supporte le calcul avec prise en compte de l'ensemble des non-linéarités localisées disponibles.

3.3.1.3 SCHEMA = 'DEVOGE' : schéma explicite d'ordre 4

Le schéma de DEVOGELAERE supporte le calcul avec prise en compte de l'ensemble des non-linéarités localisées disponibles.

3.3.1.4 SCHEMA = 'ADAPT_ORDRE2' : schéma explicite d'ordre 2

Ce schéma (appelé 'ADAPT' dans les versions antérieures du code) supporte le calcul avec prise en compte de l'ensemble des non-linéarités localisées disponibles. Cette méthode utilise le schéma des différences centrées, l'algorithme d'adaptation du pas de temps s'appuie sur le calcul d'une "fréquence apparente" :

$$f_{APt}=\frac{1}{2\pi}\sqrt{\left|\frac{\ddot{x}_t-\ddot{x}_{t-1}}{x_t-x_{t-1}}\right|}$$

On précise ci après les opérandes spécifiques à la méthode d'intégration par pas de temps adaptatifs. Ce sont les opérandes suivants du mot clé facteur INCREMENT :

◊ NB_POIN_PERIODE = N

Nombre de points par période apparente. C'est ce paramètre qui fixe la précision du calcul. Il doit être au moins égal à 20 ; sa valeur par défaut (50) garantit une précision satisfaisante (de l'ordre de 1%) dans la plupart des cas.

◊ VITE_MIN =

Méthode de calcul de la vitesse de référence utilisée pour évaluer la fréquence apparente.
Quand le dénominateur de la fréquence apparente (x_n-x_{n-1}) devient faible, celle-ci peut devenir très élevée, ce qui conduit à un raffinement injustifié du pas de temps. Pour y remédier, l'algorithme utilise le critère suivant :

$$\frac{|x_n - x_{n-1}|}{\Delta t} \leq V_{min} \Rightarrow f_{APn} = \frac{1}{2\pi}\sqrt{\left|\frac{\ddot{x}_n - \ddot{x}_{n-1}}{V_{min}\Delta t}\right|}$$

V_{min} peut être calculé de deux façons différentes selon la valeur de `VITE_MIN` :

`'NORM'` = $V_{min}(t_n) = \frac{\|V(t_n)\|}{100}$ pour tous les degrés de liberté.

Peut être utilisé :

- si le système possède plusieurs degrés de liberté,
- si l'ordre de grandeur du déplacement n'est pas trop différent selon les degrés de liberté.

`'MAXI'` = $V^i_{min}(t_n) = \frac{\underset{0<t_p<t_n}{Max}(|V^i(t_p)|)}{100}$ pour le degré de liberté i.

Peut être utilisé :

- si le système possède un petit nombre de degrés de liberté (de 1 à 3),
- pour un système à plusieurs degrés de liberté, dans le cas où l'ordre de grandeur du déplacement est très différent selon les degrés de liberté (par exemple en présence de degrés de liberté de Lagrange en sous-structuration),
- si l'ordre de grandeur de la vitesse ne varie pas trop au cours du temps.

◊ `NMAX_ITER_PAS = N`

Nombre maximal de réductions du pas de temps par pas de calcul. Il est par défaut égal à 16, ce qui limite le coefficient de réduction du pas à $0{,}75^{16} = 10^{-2}$ par itération (lorsque le pas de temps est trop élevé, on reprend le calcul avec un pas plus faible : $\Delta t_n' = 0{,}75\,\Delta t_n$).

`NMAX_ITER_PAS` peut être :

- augmenté pour permettre au pas de temps de chuter de façon plus brutale,
- diminué si le pas de temps semble excessivement raffiné, par exemple en présence de discontinuités (frottement sec, excitation discontinue, ...).

Si, à un instant donné, on atteint ce nombre maximal de réductions successives du pas de temps, alors le code va quand même considérer que le pas final est correct et passer au pas suivant. Un message d'alarme est alors émis, qui signale un éventuel risque de perte de précision et qui conseille à l'utilisateur de relancer le calcul avec des paramètres modifiés (en jouant sur `PAS`, `NMAX_ITER_PAS` et / ou `COEF_DIVI_PAS`) pour permettre de franchir la difficulté avec un pas de temps plus petit.

◊ `COEF_MULT_PAS = cmp`

Coefficient d'augmentation du pas lorsque l'erreur est suffisamment faible :

$$\Delta t_n < \frac{0{,}75}{Nf_{APn}} \Rightarrow \Delta t_{n+1} = \text{cmp}\,\Delta t_n .$$

Sa valeur par défaut ($\text{cmp} = 1.1$) garantit stabilité et précision, mais il peut en général être augmenté (au plus jusqu'à 1.3) pour accélérer l'intégration.

◊ `COEF_DIVI_PAS = cdp`

Coefficient de raffinement du pas de temps (>1) lorsque l'erreur est supérieure à 1, que le nombre d'itérations maximales (`N_MAX_ITER_PAS`) n'est pas atteint et que le pas de temps minimal n'est pas atteint :

$$\Delta t_n < \frac{1}{Nf_{APn}} \text{ , } N_{iter} < N_{iter_max} \text{ et } \Delta t_n > \text{plr}\, \Delta t_{initial}$$

$$\Rightarrow \Delta t_n = \frac{\Delta t_n}{\text{cdp}}$$

La valeur par défaut est de 1.33333334 , soit une réduction d'un facteur 0.75 .

◊ `PAS_LIMI_RELA = plr`

Coefficient appliqué au pas de temps initial pour définir la limite de raffinement et donc le pas de temps minimal :

La valeur par défaut est de 1.33333334 , soit une réduction d'un facteur 0.75 .

$$\Delta T_{min} = \text{plr}\ \Delta t_{initial}$$

3.3.1.5 SCHEMA = 'RUNGE_KUTTA_54' : schéma explicite à pas adaptatif.

Ce schéma fait partie de la famille des schémas d'intégration de type Runge-Kutta. En particulier, il s'agit du schéma d'intégration explicite de Dormand-Prince(54) [R5.06.04] à pas de temps adaptatif.
Le schéma `'RUNGE_KUTTA_54'` supporte la prise en compte de toutes les non-linéarités disponibles dans l'opérateur.

Le calcul du pas de temps optimal se fait par contrôle de l'erreur entre les approximations d'ordre 5 et 4 de la prédiction du vecteur d'état (concaténation des vecteurs de déplacement et vitesse).

Ce schéma s'appuie sur la condition de contrôle de l'erreur relative suivante :

$$err \leq tol$$

avec

$$err = \frac{1}{n} \sum_i \sqrt{\left(\frac{y_{i1} - \hat{y}_{i1}}{sc_i}\right)^2} \quad \text{et} \quad sc_i = MAX\left(|y_{i0}|, |y_{i1}|\right) + \alpha$$

où

- y_{i1} est la valeur de la prédiction d'ordre 5 de la composante i du vecteur d'état y
- $\hat{y}_{1i}$ est la valeur de la prédiction d'ordre 4 de la composante i du vecteur d'état y
- n est la taille du vecteur d'état y
- y_{0i} est la valeur de la composante i du vecteur d'état y à l'état actuel

◊ `TOLERANCE = tol`
Valeur de contrôle d'erreur relative donnée par l'utilisateur. Par défaut elle vaut `1.E-3`.

◊ `ALPHA = alpha`
Valeur de regularisation donnée par l'utilisateur intervenant dans l'expression de sc_i . Par défaut elle vaut `1.E-3`.

3.3.1.6 SCHEMA = 'RUNGE_KUTTA_32' : schéma explicite à pas adaptatif.

Comme le schéma `'RUNGE_KUTTA_54'`, le schéma `'RUNGE_KUTTA_32'` fait partie de la famille des schémas d'intégration de type Runge-Kutta. Dans ce cas, il s'agit du schéma d'intégration explicite de Bogacki-Shampine(32) [R5.06.04] à pas de temps adaptatif.
Comme le schéma précédent, il supporte la prise en compte de l'ensemble des non-linéarités disponibles dans l'opérateur.

Pour ce schéma, le calcul du pas de temps optimal se fait par contrôle de l'erreur entre les approximations d'ordre 3 et 2 de la prédiction du vecteur d'état. Le calcul du pas de temps optimal, quant à lui, se fait de manière analogue au schéma précédent.

3.3.1.7 SCHEMA = 'ADAPT_ORDRE1' : schéma explicite d'ordre 1

Ce schéma est une variante du schéma précédent 'ADAPT_ORDRE2'. C'est en fait une version du schéma d'Euler avec pas de temps adaptatif. En dehors de cette différence, ce schéma s'utilise de la même manière que le schéma adaptatif d'ordre 2 : la syntaxe des mot-clés est la même et les méthodes de pilotage du pas de temps aussi.

3.3.1.8 SCHEMA = 'ITMI' : schéma intégrale pour le calcul de la réponse de systèmes mécaniques très faiblement amortis avec couplages fluidélastiques

Ce schéma d'intégration par méthode intégrale permet, pour les systèmes faiblement amortis, d'obtenir une réponse exacte en tenant compte des variations de forces fluidélastiques obtenues en présence de chocs.
On décrit ci dessous les mots clés spécifiques au calcul de la réponse de systèmes mécaniques linéaires très faiblement amortis avec couplages fluidélastiques associés éventuellement à des non-linéarités localisées aux nœuds de type chocs et frottements.

◊ BASE_ELAS_FLUI = meles

Base modale utilisée pour le calcul.

Concept de type melasflu produit par l'opérateur CALC_FLUI_STRU [U4.66.02] qui contient l'ensemble des bases modales calculées pour les différentes vitesse d'écoulement définies. Ce mot clé est obligatoire pour la méthode 'ITMI'.

Le calcul transitoire sur base modale modifiée par le couplage fluidélastique s'effectue en prenant en compte les valeurs des amortissements ajoutés, dus à l'écoulement du fluide, qui sont présents dans le concept melasflu d'entrée . **Les amortissements modaux, récupérés de la base fluidélastique, remplacent ceux renseignés sous le mot-clé *global* AMOR_REDUIT de l'opérateur DYNA_TRAN_MODAL .**

◊ NUME_VITE_FLUI = Nvitf

Vitesse d'écoulement retenue pour le calcul (numéro d'ordre).

Permet d'extraire dans le concept melasflu la base modale correspondant à la vitesse d'écoulement retenue (cf. [U4.66.02]). Ce mot clé est obligatoire pour la méthode 'ITMI'.

◊ ETAT_STAT =

Pour les systèmes très faiblement amortis, cette option permet d'éviter un calcul coûteux de la phase linéaire précédant le premier choc. Cette phase, appelée par la suite « phase transitoire » précède l'établissement d'un régime constitué d'une succession de phases non linéaires de chocs et/ou de phases linéaires dites de « vol » selon les fonctions d'excitation du système mécanique appliquées. Le temps de transitoire correspond à un déplacement égal au jeu d'une butée. Il peut être relativement important (50 à 100 secondes).

ETAT_STAT = 'OUI' : permet le passage en un seul pas de temps de calcul de la phase transitoire.
Le passage de la phase transitoire est réalisé en supposant le système mécanique en "vol". Le temps nécessaire au passage du transitoire est estimé par l'algorithme en fonction des caractéristiques mécaniques du système en ''vol''. Cette estimation est basée sur un critère où interviennent le paramètre PREC_DUREE et les durées d'excitations dues aux efforts turbulents.

Remarque :

Si l'on demande une simulation avec calcul en un pas de temps de la phase transitoire, il

faudra veiller à introduire une durée d'excitation suffisamment longue. Cette durée doit correspondre à la durée nécessaire au passage du transitoire augmentée de la durée de simulation en régime établi souhaitée. Cette durée totale de simulation sera renseignée via les deux opérandes `INST_INIT` et `INST_FIN` sous le mot clé facteur `INCREMENT`.

`ETAT_STAT = 'NON'` : La simulation ne distingue pas l'état transitoire du régime établi.

◊ `PREC_DUREE = prec`

Permet de définir la précision choisie pour déterminer la durée de la phase transitoire selon la formule :

$T_{tr} = \frac{-\ln(prec)}{2\xi_0\omega_0}$ où ξ_0 et ω_0 désignent respectivement l'amortissement réduit et la pulsation de chaque mode considéré. La valeur par défaut de ce paramètre est 1%.

◊ `CHOC_FLUI =`

Détermine le traitement réalisé par l'algorithme pendant les phases de choc vis à vis des forces fluidélastiques.
Par défaut, la variation des forces fluidélastiques en phase de choc liée à la modification de la rigidité et de l'amortissement du système mécanique (impact sur la butée) n'est pas prise en compte.

◊ `NB_MODE = Nmode`

Nombre de modes de la base modale retenus pour le calcul dynamique.
Les modes conservés correspondent à des fréquences croissantes (premiers modes). Si `NB_MODE` n'est pas précisé, on prend tous les modes de la base modale du concept de type `melasflu`.

◊ `NB_MODE_FLUI = Nmodef`

Nombre de modes de la base modale perturbés par les phénomènes de couplage fluidélastiques en phase de choc (inférieur au nombre de modes retenus pour le calcul dynamique).
Les modes conservés correspondent aux `Nmodef` premières fréquences croissantes (premiers modes). Si `NB_MODE_FLUI` n'est pas précisé, on prend le nombre de modes retenus pour le calcul dynamique.

◊ `TS_REG_ETAB = tsimu`

Durée de la simulation souhaitée.

Dans le cas d'une simulation sans calcul préalable et en un pas de temps de la phase transitoire (`ETAT_STAT = 'NON'`), cette durée correspond à la durée de simulation quel que soit l'état du système entre les instants de début et de fin de simulation. Par conséquent on devra s'assurer que :
`TS_REG_ETAB` $\leq$ `INST_FIN - INST_INIT`

Par défaut, on aura `TS_REG_ETAB = INST_FIN - INST_INIT`

Dans le cas d'une simulation avec calcul de la phase transitoire (`ETAT_STAT = 'OUI'`), cette durée correspond à la durée de simulation réellement souhaitée lorsque la phase de chocs est établie du point de vue numérique. Par conséquent on devra s'assurer que :

`TS_REG_ETAB` $\leq$ `INST_FIN - INST_INIT` – "temps estimé transitoire"

Dans le cas où cette dernière condition n'est pas respectée, l'utilisateur en est informé avec précision du temps minimum d'excitation requis pour son calcul `INST_FIN - INST_INIT.`
Par défaut, on a : `TS_REG_ETAB = INST_FIN - INST_INIT` – "temps estimé transitoire"

Remarques :

Ce schéma d'intégration n'est pas utilisable en poursuite et ne permet pas le calcul par sous-structuration dynamique.
La présence du mot clé `CHOC` est impérative même pour des simulations de phases sans chocs dites 'phases de vol'.

3.4 Mot clé `INCREMENT`

3.4.1 Opérandes `LIST_INST / PAS / VERI_PAS / PAS_MINI / PAS_MAXI`

♦ / `LIST_INST = l_temp`
Concept liste de réels de type `listr8`.
Liste de réels définissant les instants t_i de calcul de la solution

- Schémas `'RUNGE_KUTTA_54'` et `'RUNGE_KUTTA32' :`
  Pour les schémas de type Runge-Kutta, le mot clé `LIST_INST` n'est pas pris en compte.

/ `PAS = dt`

- Schémas `'EULER', 'DEVOGE', 'NEWMARK' :`
  Pas de temps du calcul transitoire.

- Schémas `'ADAPT_ORDRE1'` et `'ADAPT_ORDRE2' :`

  Désigne le pas de temps initial utilisé par l'algorithme.
  Ce paramètre doit être suffisamment faible :

  - pour permettre le calcul des phases statiques (qui utilise toujours le pas de temps maximal),
  - pour démarrer correctement l'algorithme.

Il doit cependant être suffisamment élevé pour ne pas pénaliser l'ensemble du calcul.

- Schéma `'ITMI' :`

  Désigne le pas de temps retenu pour le premier pas de calcul (après passage éventuel du transitoire). Par la suite, l'algorithme gère automatiquement le pas de calcul en fonction de la rigidité de la structure et des zones de transition vol/choc.

- Schémas `'RUNGE_KUTTA_54'` et `'RUNGE_KUTTA32' :`

  Désigne le pas de temps initial proposé par l'utilisateur. Si l'erreur de prédiction entre les ordres de contrôle vérifie $err \leq 1$, alors il s'agit du premier pas de calcul. Sinon, l'algorithme choisit automatiquement le pas de temps nécessaire afin de vérifier cette condition. Par la suite, le choix du pas de temps dans les algorithmes de Runge-Kutta est géré automatiquement.

◊ `VERI_PAS = rep`

Vérification du pas de temps de calcul relativement au pas de temps limite déterminé en fonction de la fréquence la plus élevée des modes de la base modale considérée ou des bases des sous-structures.

Opérandes spécifiques à une intégration par pas de temps adaptatifs avec les schémas `'ADAPT_ORDRE1'`, `'ADAPT_ORDRE2'` ainsi que `'RUNGE_KUTTA_54'` et `'RUNGE_KUTTA_32'` .

◊ `PAS_MAXI = dtmax`

Valeur maximale du pas de temps. Si les conditions d'augmentation du pas de temps sont remplies, le pas de temps courant pourra alors augmenter jusqu'à cette valeur limite.

Si l'utilisateur ne donne pas de valeur à ce paramètre facultatif, les schémas `'ADAPT_ORDRE1'`, `'ADAPT_ORDRE2'` estimeront une valeur notée dts à partir de la fréquence de coupure de la base (éventuellement corrigée par les raideurs de chocs). En revanche, les schémas de Runge-Kutta n'auront aucune limitation supérieur en termes de pas de temps.

Pour retrouver le fonctionnement des versions antérieures du code, il suffit d'imposer : $dtmax = dt$, donc la même valeur au paramètre `PAS` qu'à `PAS_MAXI`.
Si l'utilisateur donne une valeur supérieure à dts, une alarme sera émise prévenant d'un risque de perte de précision.

◊ `PAS_MINI = dtmin`

Valeur minimale du pas de temps. Si les conditions de diminution du pas de temps sont remplies, le pas de temps courant pourra alors diminuer jusqu'à cette valeur limite.
Si l'utilisateur ne donne pas de valeur à ce paramètre facultatif, alors le code calculera le pas de temps minimal à partir de `PAS_LIMI_RELA`.
Pour retrouver le fonctionnement des versions antérieures du code, il suffit donc de ne pas définir `PAS_MINI`.

3.4.2 Opérandes INST_INIT / INST_FIN / NUME_FIN

◊ `INST_INIT = to`

- Méthodes `'EULER'`, `'DEVOGE'`, `'NEWMARK'`, `'ADAPT_ORDRE1'` et `'ADAPT_ORDRE2'` :

  Instant de début du calcul transitoire. En cas de reprise, on utilise le mot clé `ETAT_INIT` : sous ce mot clé, l'instant initial est récupéré avec l'opérande `INST_INIT` ou pris égal au dernier instant de calcul précédent archivé. L'opérande `INST_INIT` doit donc être utilisé uniquement s'il n'y a pas de reprise d'un calcul précédent.

- Schéma `'ITMI'` :

  Désigne l'instant de début de simulation. Lorsque le calcul en un pas de temps de la phase transitoire est demandé, le simulation débute à `INS_INIT` + « temps de calcul du transitoire »

◊ `/ INST_FIN = tf`

Instant de fin de la simulation.

`/ NUME_FIN = tf`

Numéro de l'instant de fin de calcul dans `LIST_INST`

3.5 Mot clé ETAT_INIT

Mot clé facteur qui permet une poursuite d'un calcul transitoire, en prenant comme état initial :

- soit un résultat issu d'un calcul par synthèse modale précédent `EXCIT(RESULTAT)` ;
- soit des déplacements et vitesses exprimés sous forme de vecteurs assemblés généralisés `EXCIT(DEPL` et `VITE)`

Remarques :

- *Cette fonctionnalité n'est pas disponible pour un calcul par sous-structuration transitoire sans double projection ni pour la méthode ITMI.*
- *Lors d'une poursuite, l'état d'adhérence ou de choc n'est pas sauvegardé.*
- *Les déplacements et vitesses généralisés doivent être établis par l'opérateur `PROJ_VECT_BASE` [U4.63.13] à partir de la base modale utilisée pour les matrices de rigidité généralisées ou par l'opérateur `RECU_GENE` [U4.71.03] appliqué à un calcul précédent.*

3.5.1 Opérandes RESU / DEPL / VITE

♦/ RESULTAT = tran

Concept de type tran_gene issu d'un calcul précédent avec DYNA_TRAN_MODAL.

/ | DEPL = do

Concept de type vect_asse_gene, déplacements généralisés initiaux.

| VITE = vo

Concept de type vect_asse_gene, vitesses généralisées initiales.

3.5.2 Opérandes INST_INIT / NUME_ORDR

◊ / INST_INIT = to

Instant du calcul précédent à extraire et à prendre comme état initial dans le cas d'une reprise. En l'absence de cet opérande, l'instant de reprise est pris égal au dernier instant de calcul précédent archivé.

/ NUME_ORDRE = nuord

Désigne le numéro d'archivage du calcul précédent à extraire et à prendre comme état initial dans le cas d'une reprise.

3.5.3 Opérande CRITERE

◊ CRITERE

Indique avec quelle précision la recherche de l'instant doit se faire :

'RELATIF' : intervalle de recherche [(1-prec).instant, (1+prec).instant]
'ABSOLU' : intervalle de recherche [instant-prec, instant+prec]

Le critère est 'RELATIF' par défaut.

3.5.4 Opérande PRECISION

```
◊ PRECISION =  /  1.E-06    [DEFAUT]
               /  prec      [R8]
```

Indique avec quelle précision la recherche de l'instant doit se faire.

3.6 Description du chargement sous variables séparées : mot clé EXCIT

◊ EXCIT

Mot clé définissant le chargement. Ce mot clé doit être répété autant de fois qu'il y a de vecteurs chargement généralisé f_i. Le chargement total est la somme de ces vecteurs chargement. Ce mot-clé permet de définir le chargement sous a forme de vecteurs généralisés multipliés par des fonctions multiplicatrices.

3.6.1 Opérandes VECT_ASSE_GENE / NUME_ORDRE

Le chargement est pris en compte sous forme de vecteur projeté sur la base modale EXCIT=_F(VECT_ASSE_GENE) ou sous forme de composante modale EXCIT =_F(NUME_MODE) ou les deux à la fois.

˜ VECT_ASSE_GENE = v

Vecteur généralisé permettant de décrire la répartition spatiale du chargement.
Concept de type vect_asse_gene.

Les vecteurs généralisés doivent être établis par l'opérateur PROJ_VECT_BASE [U4.63.13] à partir de la base modale utilisée pour les matrices généralisées. Dans le cas d'un calcul par sous-structuration dynamique, les vecteurs généralisés doivent être établis par l'opérateur ASSE_VECT_GENE [U4.65.05] à partir de la numérotation généralisée utilisée pour les matrices généralisées.

/ NUME_ORDRE = nmordr

Numéro d'ordre du mode d'excitation de la structure (Attention ! Il ne faut pas confondre le numéro d'ordre du mode – donné par le calcul modal dans l'ordre où ils ont été calculés – et le numéro du mode, intitulé dans *Code_Aster* NUME_MODE).

3.6.2 Opérande FONC_MULT / COEF_MULT

♦ / FONC_MULT = f

Fonction du temps (fonction) permettant de décrire l'évolution temporelle du vecteur chargement.

/ COEF_MULT = a

Coefficient multiplicateur du vecteur généralisé (valeur réelle constante par rapport au temps).

3.7 Mot clé EXCIT_RESU

Mot clé permettant de définir un chargement sous la forme d'une évolution temporelle généralisée, sans séparation de variables (cas le plus général). Cette évolution temporelle peut avoir été calculée à partir de l'opérateur PROJ_BASE, option RESU_GENE, qui réalise la projection d'un résultat dynamique transitoire (dyna_trans).

3.7.1 Mot-clé RESULTAT

♦ / RESULTAT = resu_gene

Structure de données resu_gene définissant le chargement généralisé.

3.7.2 Opérande COEF_MULT

◊ / COEF_MULT = f

Coefficient multiplicateur, vaut 1.0 par défaut.

3.8 Cas particulier de l'analyse sismique

3.8.1 Prise en compte des modes négligés par correction statique : mots clés CORR_STAT, MODE_CORR et D_FONC_*

Lors de l'analyse sismique d'une structure mono excitée, il est possible de prendre en compte, a posteriori, l'effet statique des modes négligés. Dans ce cas, lors du retour sur la base physique, les déplacements relatifs calculés (respectivement les vitesses et accélérations relatives) sont corrigés par un pseudo-mode.
On trouvera les détails de ce type de correction dans [R4.05.01].

À l'intérieur du mot clé facteur EXCIT, CORR_STAT='OUI' permet la prise en compte des modes négligés par correction statique, il est alors obligatoire de renseigner les mots clés MODE_CORR, D_FONC_DT et D_FONC_DT2.

◊ MODE_CORR = modcor

Concept de type mult_elas produit par la macro-commande `MACRO_ELAS_MULT` [U4.51.02] ou mode_meca qui correspond à la réponse statique linéaire de la structure à un chargement unitaire de type force imposée (accélération uniforme) dans la direction du séisme considérée.
On note qu'il y a autant de cas de charge que de direction de séisme.

◊ `EXCIT = _F( CORR_STAT )`

Si `MODE_CORR` est présent, `CORR_STAT='OUI'` permet de prendre en compte la contribution de la correction modale a posteriori pour chaque occurrence du mot clé `EXCIT`.

◊ `EXCIT = _ F( D_FONC_DT et D_FONC_DT2 )`

`D_FONC_DT` et `D_FONC_DT2` sont respectivement les dérivées premières et dérivées secondes du temps de l'accélérogramme défini, dans chaque direction sismique considérée, par l'opérande `FONC_MULT`. Elles pondèrent la contribution de la correction modale a posteriori pour chaque occurrence du mot clé `EXCIT` afin d'obtenir respectivement les corrections de vitesse et d'accélération sur la base physique.

Remarques :

- *La prise en compte de la correction statique exclue celle du multi-appuis.*
- *Le concept* `mult_elas` *doit s'appuyer sur une numérotation des équations cohérente (même profil et même option de renumérotation) avec celle du système résolu dans l'opérateur* `DYNA_TRAN_MODAL`*.*
- *A la* i *ème occurrence du mot clé* `EXCIT` *correspond la* i *ème solution élastique de* `MODCOR` *.*
- *Pour que la correction statique soit effectivement prise en compte lors du retour vers les coordonnées physiques dans l'opérateur* `REST_GENE_PHYS` *ou* `RECU_FONCTION` *il faut préciser* `CORR_STAT_='OUI'`.

3.8.2 Prise en compte du multi-appuis : mots clés MODE_STAT, MULTI_APPUI et ACCE, VITE, DEPL

Dans le cas d'une structure multi-supportée, afin de restituer les grandeurs calculées dans le repère absolu ou prendre en compte des non linéarités localisées, il faut calculer la réponse généralisée en prenant en compte la composante d'entraînement.
Pour plus de détails, on se reportera à la référence [R4.05.01].

Les mots clés `MODE_STAT`, `MULT_APPUI`, `ACCE`, `VITE`, `DEPL`, `DIRECTION` et `NOEUD` ou `GROUP_NO`) spécifiques à la prise en compte du caractère multi-supporté doivent être simultanément présents.
Un `cham_no` issu de la projection d'un `CALC_CHAR_SEISME` représente le vecteur d'excitation sur l'appui. Il ne doit pas être oublié, même si l'information peut paraître redondante avec la donnée de l'appui et de la direction de séisme.

◊ `MODE_STAT = psi`

Concept de type `mode_meca` produit par la commande `MODE_STATIQUE` [U4.52.14] qui correspond aux (3 ou 6) `nb_supports` modes statiques (où `nb_supports` est le nombre de supports qui subissent une accélération différente).

◊ `EXCIT = _F( MULT_APPUI )`

Si on calcule la réponse sismique d'une structure multi-supportée, `MULT_APPUI = 'OUI'`, on compare à chaque instant, le vecteur des déplacements absolus de chacun des points de choc considérés, afin de déterminer si il y a choc et de calculer les forces de choc correspondantes.
Sinon, `MULT_APPUI = 'NON'`, on compare à chaque instant, le vecteur des déplacements relatifs de chacun des nœuds susceptibles de choquer.

```
◊ EXCIT = _F( /  ◊  ACCE = ac,
                 ◊  VITE = vi,
                 ◊  DEPL = dp )
```

Noms des fonctions accélération (ACCE), vitesse (VITE) et déplacement (DEPL) imposées lors du calcul de la réponse sismique de structures multi-supportées.

Remarque :

> *Si la structure est mono-excitée, l'accélérogramme est défini par le mot clé FONC_MULT .*

◊ EXCIT = _F(DIRECTION =(dx, dy, dz, drx, dry, drz))

Composantes du vecteur donnant la direction du séisme dans le repère global.

```
◊ EXCIT = _F(  /   NOEUD       =     lno
               /   GROUP_NO    =     lgrno )
```

Liste des noms de nœuds (ou de groupe de nœuds) correspondants aux appuis concernés où le séisme est imposé.

◊ EXCIT = _F(VECT_ASSE_GENE = v)

Vecteur projeté de l'excitation sismique (issu de CALC_CHAR_SEISME [U4.63.01])

3.9 Prise en compte d'un transitoire de vitesse de rotation

3.9.1 Opérande VITESSE_VARIABLE

Précise si la vitesse de rotation du rotor est variable en fonction du temps (VITESSE_VARIABLE = 'OUI' pour les transitoires de vitesse) ou constante (VITESSE_VARIABLE = 'NON').

3.9.2 Opérandes VITE_ROTA, MATR_GYRO, ACCE_ROTA et MATR_RIGY

Ces opérandes sont les paramètres définissant le transitoire de vitesse de rotation.

Si VITESSE_VARIABLE = 'OUI', alors il faut renseigner les paramètres suivants :

♦ VITE _ROTA = fonction donnant la loi de vitesse angulaire imposée au rotor
♦ MATR _GYRO = matrice d'amortissement gyroscopique
◊ ACCE _ROTA = fonction donnant la loi d'accélération angulaire imposée au rotor
◊ MATR _RIGY = matrice de raideur gyroscopique

Remarque :

> *Si ACCE _ROTA n'est pas fournie, elle est négligée (ie. pas de dérivation numérique à partir de VITE _ROTA).*

Si VITESSE_VARIABLE = 'NON', il faut renseigner la valeur de la vitesse de rotation constante.

♦ VITE_ROTA = vit esse de rotation de l'arbre [par défaut 0.0]

3.10 Prise en compte de non linéarités localisées de type choc, frottement, et rotor fissuré

3.10.1 Non linéarités localisées de type choc et frottement : mot clé CHOC

◊ CHOC

Ce mot clé facteur est utilisé pour l'étude de la réponse de structures (généralement élancées) dont les déplacements sont limités en un (ou plusieurs) point(s) -précisés a priori par l'utilisateur- par la présence d'un obstacle (les différents types d'obstacles disponibles sont décrit dans la documentation [U4.44.21] de l'opérateur DEFI_OBSTACLE), d'une autre structure antagoniste.

3.10.1.1 Opérande INTITULE

◊ INTITULE = int

Intitulé (huit caractères au maximum) permettant de nommer la non-linéarité. Si rien n'est précisé par l'utilisateur, l'intitulé est le nom du NOEUD_1.

3.10.1.2 Opérandes NOEUD_1 / NOEUD_2 / GROUP_NO_1 / GROUP_NO_2 . / GROUP_MA

♦ NOEUD_1 ou GROUP_NO_1
Nœud ou nom du groupe de nœud de la structure sur lequel porte la condition de non-linéarité.
Dans le cas d'un calcul non-linéaire par sous-structuration dynamique, on indique sous ce mot clé le nœud de choc appartenant à la première sous-structure (les différentes sous-structures n'appartiennent pas au même maillage).

◊ NOEUD_2 ou GROUP_NO_2
Nœud ou nom du groupe de nœud de la seconde structure sur lequel porte la condition de non-linéarité. Cette opérande est spécifique à la définition d'un contact entre deux structures mobiles.
Dans le cas d'un calcul non-linéaire par sous-structuration dynamique, on précise le nœud de choc coïncidant avec le nœud indiqué dans NOEUD_1 (ou GROUP_NO_1), mais appartenant à la deuxième sous-structure.

Remarque :

On vérifie que les groupes de nœuds contiennent bien un et un seul nœud.

♦ GROUP_MA ou MAILLE
On peut aussi entrer les nœuds de chocs en vis à vis sous la forme de mailles SEG2 dessinées dans le maillage. Ainsi on conserve une même façon de décrire les chocs que pour DYNA_NON_LINE avec les éléments discrets de choc (DIS_CHOC). On peut entrer une liste de MAILLE ou de GROUP_MA.

3.10.1.3 Opérande OBSTACLE

♦ OBSTACLE = obs
Nom du concept de type obstacle définissant la géométrie d'un obstacle indéformable ou la forme enveloppe du jeu entre deux structures antagonistes. Il est produit par l'opérateur DEFI_OBSTACLE [U4.44.21].

3.10.1.4 Opérande NORM_OBST

♦ NORM_OBST = nor
Liste de 3 réels définissant la normale au plan de coupe de l'obstacle, c'est-à-dire le vecteur X_{loc} . On conseille que X_{loc} soit la direction de la fibre neutre ou d'une génératrice de la structure étudiée.

3.10.1.5 Opérande ORIG_OBST

◊ ORIG_OBST = ori
Liste de 3 réels définissant la position de l'origine de l'obstacle dans le repère global (mot clé obligatoire dans le cas de chocs entre une structure mobile et une paroi fixe). Dans le cas de chocs entre deux structures mobiles, le code considère par défaut que l'origine est située au milieu des deux nœuds de choc NOEUD_1 (ou nœud du GROUP_NO_1) et NOEUD_2 (ou nœud du GROUP_NO_2).

3.10.1.6 Opérande JEU

◊ JEU = jeu
Dans le cas d'un choc entre une structure mobile et un obstacle indéformable, l'opérande JEU représente :
- la demi-distance inter-plans pour des obstacles de type PLAN_Y et PLAN_Z
- le rayon de l'obstacle circulaire pour un obstacle de type CERCLE

Ce mot clé est inutilisé dans le cas d'obstacles discrétisés par segments de type DISCRET.

Remarque :

L'obstacle de type `PLAN_Y` ou `PLAN_Z` comporte en fait deux obstacles plans. Ainsi dans le cas où l'utilisateur souhaite modéliser le choc sur un plan unique, pour ne pas être gêné par le rebond de la structure étudiée sur le plan symétrique, on conseille à l'utilisateur de le repousser très loin (cf. [Figure 3.6.1.6-a]), j représente le jeu réel entre la structure étudiée et l'obstacle .

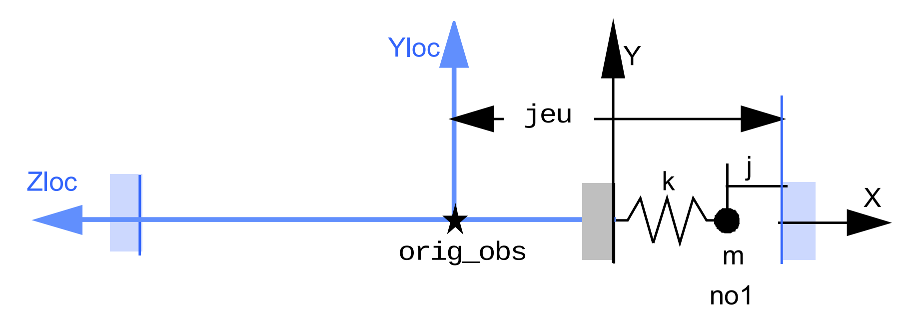


Figure 3.6.1.6-a : Système masse-ressort impactant une paroi fixe

Remarque :

Le mot-clé `JEU` n'est pas utilisé dans le cas de choc entre structures mobiles.

Les différents cas de jeux sont représentés dans la documentation de `DEFI_OBSTACLE` [U4.44.21].

3.10.1.7 Opérande `ANGL_VRIL`

◊ `ANGL_VRIL = gamma`

γ , angle en degrés définissant la position angulaire du repère local de l'obstacle dans son plan.

Par convention, la normale n au plan de coupe de l'obstacle, `NORM_OBST` définit l'axe X_{loc} du repère local. On passe du repère global X, Y, Z au repère du plan de l'obstacle n, y_2, z_2 par un produit de deux rotations d'angles α autour de Z puis β autour du transformé y_1 de Y. La position de l'obstacle dans ce plan est obtenue par une rotation d'angle β autour de la direction normale X_{loc} (cf. [Figure 3.6.1.7-a]).

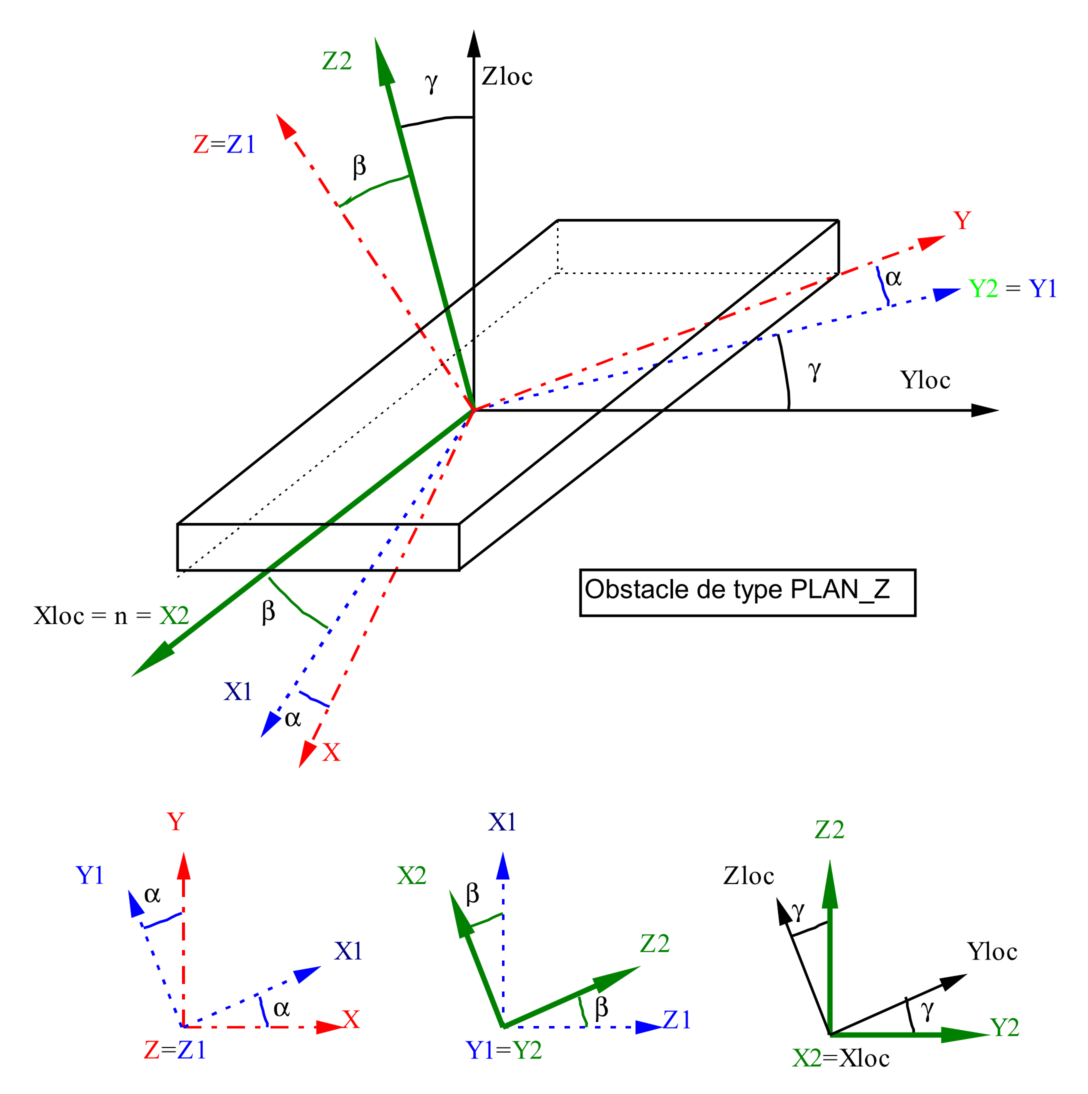


Figure 3.6.1.7-a : Rotations permettant de passer du repère global au repère local de l'obstacle.

Les angles α et β sont déterminés automatiquement à partir de la normale à l'obstacle n . Le repère local X_{loc}, Y_{loc}, Z_{lo} se déduit ensuite du repère n, y_2, z_2 par rotation d'un angle de vrille ANGL_VRIL autour de n .

Remarque :

- *Si l'utilisateur ne précise rien, l'angle de vrille est calculé par le code dans le cas de chocs entre structures mobiles avec des obstacles de type BI_PLAN .*
- *En ce qui concerne les autres types d'obstacles, la valeur par défaut de gamma est zéro.*

3.10.1.8 Opérandes DIST_1 / DIST_2

◊ DIST_1 = dist1

Distance caractéristique de matière entourant NOEUD_1 : no1 (ou GROUP_NO_1). Opérande spécifique au contact entre deux structures mobiles.

◊ DIST_2 = dist2

Distance caractéristique de matière entourant NOEUD_2 : no2 (ou GROUP_NO_2). Opérande spécifique au contact entre deux structures mobiles.

Remarques :

- *DIST_1 et DIST_2 sont définies au sens des normales sortantes des deux solides en vis-à-vis (DIST_1 et DIST_2 sont positives car elles représentent l'épaisseur des structures étudiées).*
- *Du fait du calcul de la distance normale de choc, la somme de DIST_1 et de DIST_2 doit être suffisamment grande par rapport à l'amplitude supposée du déplacement relatif des nœuds de chocs (cf. [R5.06.03]).*

3.10.1.9 Opérandes SOUS_STRUC_1 / SOUS_STRUC_2

◊ SOUS_STRUC_1 = ss1

Nom de la sous-structure qui contient le nœud de choc renseignant le mot clé NOEUD_1 (ou GROUP_NO_1).

◊ SOUS_STRUC_2 = ss2

Nom de la sous-structure qui contient le nœud de choc renseignant le mot clé NOEUD_2 (ou GROUP_NO_2).

3.10.1.10 Opérande REPERE

◊ REPERE = rep

Précise le repère dans lequel la position de l'obstacle est définie.

/ 'GLOBAL'

La position absolue de l'obstacle est définie indépendamment des rotations et translations auxquelles sont soumises les différentes sous-structures.

/ nom_sst

Nom d'une sous-structure.
La position et la normale de l'obstacle sont déterminées dans le repère utilisé pour définir les coordonnées des nœuds de la sous-structure nom_sst, la position et la normale finales de l'obstacle étant le résultat de la rotation et de la translation auxquelles est soumise la sous-structure.

3.10.1.11 Opérande RIGI_NOR

♦ RIGI_NOR = kn
Valeur de la rigidité normale de choc (unité N/m en USI).

3.10.1.12 Opérande AMOR_NOR

◊ AMOR_NOR = cn
Valeur de l'amortissement normal de choc (unité $N\,m/s$ en USI).

3.10.1.13 Opérande RIGI_TAN

♦ RIGI_TAN = kt
Valeur de la rigidité tangentielle de choc (unité N/m en USI).

3.10.1.14 Opérande AMOR_TAN

◊ AMOR_TAN = ct
Valeur de l'amortissement tangentiel de choc (unité $N\,m/s$ en USI).

Remarque :

Si une raideur k_t est spécifiée et que le mot clé AMOR_TAN est absent, le code calcule un amortissement optimisé de façon à minimiser les oscillations résiduelles en adhérence selon la formule :

$$c_t = 2\sqrt{(k_i + k_t) m_i} - 2\xi_i \sqrt{k_i m_i}$$

où i est l'indice du mode prépondérant dans la réponse de la structure.

3.10.1.15 Opérande FROTTEMENT

◊ `FROTTEMENT = / 'NON'`

La condition de contact est sans frottement.

`/ 'COULOMB'`

♦ `COULOMB = mu`

Valeur du coefficient de frottement (sans dimension).

`/ 'COULOMB_STAT_DYNA'`

♦ `COULOMB_STAT = mus`

Valeur du coefficient d'adhérence (sans dimension).

♦ `COULOMB_DYNA = mud`

Valeur du coefficient de glissement (sans dimension).

3.10.2 Non linéarités localisées de rotor fissuré : ROTOR_FISS

Les opérandes suivants sont spécifiques au calcul transitoire avec non-linéarité localisée de type « rotor fissuré » pour les calculs de ligne d'arbre modélisée en 1D (poutre). La fissure est considérée totalement incluse dans une section du rotor. Elle est délimitée par deux nœuds distincts mais de coordonnées confondues, l'un reliée à la partie gauche de la ligne d'arbre, l'autre à la partie droite. Ils représentent respectivement la lèvre gauche et la lèvre droite de la fissure.
Le comportement de la fissure est donné par une loi de raideur de fissure et sa dérivée. Cette loi est déterminée par ailleurs par des calculs 3D en quasi-statique. Elle ne dépend pas de la géométrie du rotor mais uniquement de la forme de la fissure et d'un coefficient de dimension.

◊ `ROTOR_FISS=_F(`

/ ♦ `/NOEUD_G` = lèvre gauche de la fissure

`/GROUP_NO_G` = groupe de nœud nommant la lèvre gauche de la fissure

♦ `/NOEUD _D` = lèvre droite de la fissure

`/GROUP_NO_D` = groupe de nœud nommant la lèvre gauche de la fissure

♦ `ANGL_INIT` = angle initial du fond de fissure par rapport à sa définition dans la loi de comportement de fissure [par défaut 0.0]

◊ `ANGL_ROTA` = fonction donnant la loi imposée de position angulaire du fond de fissure par rapport à sa définition dans la loi de comportement de fissure (transitoires de vitesse)

♦ `K_PHI` = loi de comportement en raideur de la fissure

♦ `DK_DPHI` = dérivée de la loi de comportement en raideur

)

Orientation de l'axe du rotor :
Pour respecter le sens de rotation trigonométrique, il est important de bien orienter le rotor : l'axe du rotor est automatiquement orienté par la fissure, en allant du bord gauche vers le bord droite de la fissure.

3.11 Mot clé VERI_CHOC

Mot clé qui permet d'évaluer a posteriori, l'aptitude de la base modale à représenter correctement les impacts.

Si `VERI_CHOC` est présent, on calcule en chaque nœud de choc et pour chaque mode, le taux de reconstitution de la solution statique : $t_s = K_{statique} \sum_{i=1}^{n} \frac{({}^T\Phi_i . F_{impo})^2}{k_i}$ et, pour information, le taux de

reconstitution de l'effort tranchant : $t_N = \sum_{i=1}^{n} \frac{{}^T\Phi_i . F_{impo}}{k_i} . ({}^T F_{impo} . K . \Phi_i)$. On calcule ensuite les valeurs cumulées sur l'ensemble des modes qui constituent la base modale utilisée.

On vérifie que le rapport de la souplesse négligée (souplesse statique moins souplesse statique reconstituée) sur la souplesse de choc reste inférieur à la valeur donnée par l'opérande `SEUIL` (`SEUIL` vaut 0.5 par défaut) sinon :

- si `STOP_CRITERE = 'OUI'` on arrête l'exécution du programme (c'est le cas par défaut) ;
- si `STOP_CRITERE = 'NON'` on continue l'exécution du programme avec émission d'une alarme.

Remarques :

- *Cette fonctionnalité n'est disponible que pour des obstacles de type `plan` ou `bi_plan` .*
- *Si le taux de reconstitution de la solution statique est inférieur à la valeur du seuil, on conseille à l'utilisateur de compléter la base modale par les modes locaux aux points de choc qui ont une souplesse locale importante.*
- *La formule n'est pas applicable en cas de modes statiques (matrice de rigidité non inversible). Le calcul se poursuit alors sans vérification des critères de choc et l'utilisateur en est averti.*

3.12 Mot clé `ANTI_SISM`

Le mot clé `ANTI_SISM` est incompatible avec un calcul par sous-structuration dynamique. Il permet de calculer la force non linéaire qui existe si un dispositif anti-sismique est placé entre les deux nœuds antagonistes dont les noms sont précisés par les mots clés (`NOEUD_1` ou `GROUP_NO_1` et `NOEUD_2` ou `GROUP_NO_2`) :

$$F_D = K_2 x + \frac{(K_1 - K_2) x}{\sqrt{1 + \left(K_1 \frac{x}{P_y}\right)^2}} + C\, sign(\dot{x}) \left| \dot{x} \frac{x}{x_{max}} \right|^{\alpha}$$

◊ `RIGI_K1, RIGI_K2, SEUIL_FX, C, PUIS_ALPHA` et `DX_MAX`

Paramètres de la force due à la présence d'un dispositif anti-sismique.

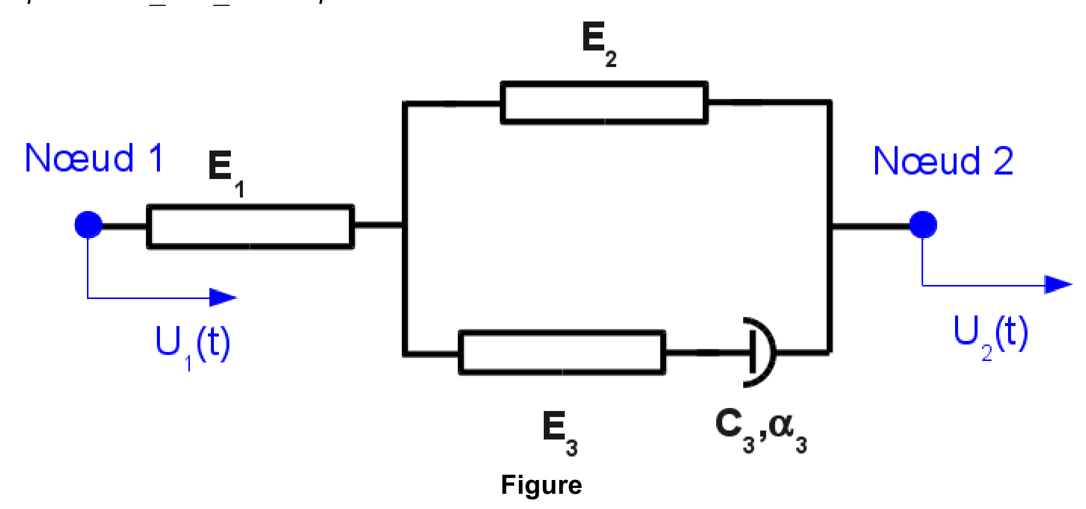
A titre d'exemple, les valeurs des paramètres pour un dispositif anti-sismique de type JARRET sont :

$K1 = 6.E+06\,N/m$, $K2 = 0.53E+06\,N/m$, $Py = 1200.$, $C = 0.07E+05\,Nm/s$, $alpha = 0.2$ et $xmax = 0.03\,m$ (si le problème est posé en USI).

3.13 Mot clé `DIS_VISC`

C'est un comportement viscoélastique non linéaire entre deux nœuds, *cf.* [R5.03.17]. Ce comportement n'affecte que le degré de liberté DX local de l'élément. La direction x locale de l'élément va du nœud 1 au nœud 2.

Remarque : Les résultats concernant l'effort, les déplacements visqueux et relatifs entre les deux nœuds, ainsi que la dissipation du dispositif non-linéaire peuvent être sauvegardés dans un fichier directement exploitable par les commandes de Code_Aster . La définition du fichier se fait par le mot clef simple `UNITE_DIS_VISC` *qui est sous le mot clef facteur* `IMPRESSION` *de la commande.*



Figure

3.13.1 Syntaxe

```
◊     DIS_VISC = _F(
          ♦   / NOEUD_1          = no1,                    [noeud]
              / GROUP_NO_1       = grno1,               [group_no]
          ♦   / NOEUD_2          = no2,                    [noeud]
              / GROUP_NO_2       = grno2,               [group_no]

          ♦   / K1               = k1,                         [R]
              / UNSUR_K1         = usk1,                       [R]
          ♦   / K2               = k2,                         [R]
              / UNSUR_K2         = usk2,                       [R]
          ♦   / K3               = k3,                         [R]
              / UNSUR_K3         = usk3,                       [R]
          ♦   C                  = c,                          [R]
          ♦   PUIS_ALPHA         = / 0.5                  [defaut]
                                   / alpha,                    [R]

          ◊   ITER_INTE_MAXI     = / 20                   [defaut]
                                   / iter                      [I]
          ◊   RESI_INTE_RELA     = / 1.0E-06              [defaut]
                                   / resi                      [R]
      )
```

3.13.2 Opérandes liés à la position du dispositif

♦ `NOEUD_1` ou `GROUP_NO_1`
♦ `NOEUD_2` ou `GROUP_NO_2`

Nœuds ou nom des groupes de nœud de la structure entre lesquels le dispositif non-linéaire est placé. Si on utilise un `GROUP_NO` pour définir l'un des points d'accrochage du dispositif, le groupe de nœud ne doit contenir qu'un seul nœud.
Lors du calcul, il est nécessaire de connaître la direction du dispositif non-linéaire car il ne fonctionne que dans son axe. Il faut donc que la distance entre les deux nœuds soit non nulle.

3.13.3 Opérandes liés au comportement

Le comportement DIS_VISC est un comportement rhéologique viscoélastique non linéaire, de type Zener étendu, permettant de schématiser le comportement d'un amortisseur uni-axial, entre deux nœuds.

Pour la direction locale x (et seulement celle-là) du dispositif, on fournit cinq coefficients. Leurs unités doivent être en accord avec l'unité des efforts, l'unité des longueurs et l'unité de temps du problème :

- K1 : raideur élastique de l'élément 1 du modèle rhéologique,
- K2 : raideur élastique de l'élément 2 du modèle rhéologique,
- K3 : raideur élastique de l'élément 3 du modèle rhéologique,

- UNSUR_K1 : souplesse élastique de l'élément 1 du modèle rhéologique,
- UNSUR_K2 : souplesse élastique de l'élément 2 du modèle rhéologique,
- UNSUR_K3 : souplesse élastique de l'élément 3 du modèle rhéologique,

- PUIS_ALPHA : puissance du comportement visqueux de l'élément α ,
- C : coefficient du comportement visqueux de l'élément.

Il existe des conditions à respecter sur les valeurs des coefficients pour que la tangente soit toujours définie :

$k1 \geq 10^{-8}$ $usk1 \geq 0$ $k3 \geq 10^{-8}$ $usk3 \geq 0$ $usk2 \geq 10^{-8}$ $k2 \geq 0$ $C \geq 10^{-8}$
$10^{-8} \leq \alpha \leq 1$

On ne peut donc pas avoir à la fois $usk1=0$, $usk3=0$ et $k2=0$ c'est-à-dire le cas de l'amortisseur seul.

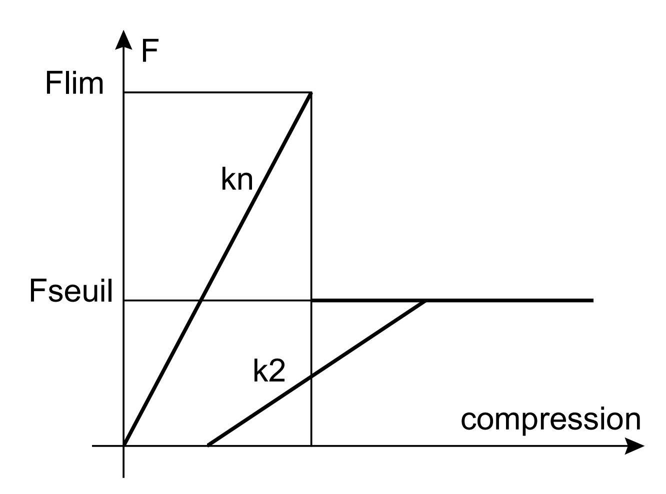
3.13.4 Opérandes liés à la convergence du comportement du dispositif

```
◊   ITER_INTE_MAXI    = / 20                         [defaut]
                        / iter                            [I]
◊   RESI_INTE_RELA    = / 1.0E-06                    [defaut]
                        / resi                            [R]
```

Ces opérandes ont la même signification que lorsqu'ils sont utilisés avec la commande STAT_NON_LINE/COMPORTEMENT [U4.51.11].
La relation de comportement DIS_VISC nécessite de résoudre un système non linéaire par une méthode de Runge-Kutta d'ordre 5 à pas adaptatif. Le contrôle de l'algorithme (nombre d'itération et résidu) sont utilisés pour tester la convergence et adapter le pas si besoin.

3.14 Mot clé FLAMBAGE

Ce mot clé est utilisé pour la détection de flambage éventuel et pour l'évaluation de la déformation résiduelle d'un élément lors d'un choc entre deux structures mobiles ou entre une structure mobile et une paroi fixe. La force de réaction lors d'un choc avec prise en compte du flambage peut être résumée par le schéma suivant :



On considère qu'il y a flambage si la force de réaction F atteint la valeur limite F_{lim} définie par l'utilisateur. La rigidité normale de choc après flambage $k2$ est ensuite différente de la rigidité avant flambage kn .

Seuls les opérandes spécifiques au mot clé `FLAMBAGE` sont détaillés. Les autres mots clés permettent de définir les lieux de choc et sont identiques aux opérandes du mot clé `CHOC`.

◊ `FNOR_CRIT = flim`

Force normale limite qui entraîne le flambage de la structure.

◊ `FNOR_POST_FL = fseuil`
Force normale limite après flambage qui provoque une déformation résiduelle de la structure.

◊ `RIGI_NOR_POST_FL = k2`
Valeur de la rigidité normale après flambage.

3.15 Mot clé RELA_EFFO_DEPL

◊ `RELA_EFFO_DEPL`
Mot clé facteur permettant de définir une relation force-déplacement ou moment-rotation sur un degré de liberté donné sous la forme d'une courbe non linéaire.

3.15.1 Opérande NOEUD

♦ `NOEUD = no`
Nom du nœud de la structure sur lequel porte la relation.

3.15.2 Opérande SOUS_STRUC

◊ `SOUS_STRUC = ss`
Nom de la sous-structure contenant le nœud renseignant l'opérande `NOEUD`.

3.15.3 Opérande NOM_CMP

◊ `NOM_CMP = nomcmp`
Nom de la composante du nœud de la structure sur laquelle porte la relation.

3.15.4 Opérande RELATION

♦ `RELATION = f`
Nom de la fonction non linéaire.

La relation non linéaire doit être définie sur *sur* $]-\infty,\infty[$. La phase non-linéaire dans les post-traitements correspond à la plage d'instants quand la relation non linéaire était non-nulle.

L’équation d’équilibre, pour une structure soumise à une accélération de sol horizontale a_x dans la direction x , et ayant des termes de correction provenant de non-linéarités, s’écrit :

$$M\ddot{x}+C\dot{x}+Kx=-M a_x+F_c$$

où F_c est la force corrective due à la non linéarité du sol. Elle peut être, par exemple, définie par la relation suivante (cf. cas test SDND103) :

$$F_c=kx-f(x)$$

avec :

si $x\geq x_0$, $f(x)=k\left(\frac{|x|}{x_0}\right)x$.

Dans l’exemple ci dessus, on impose donc, sous l’opérande `RELATION` la fonction :

$$F_c(x)=\frac{k}{x_0}x[|x|-x_0] \quad \text{pour} \quad |x|>x_0$$
$$F_c(x)=0 \quad \text{pour} \quad |x|\leq x_0$$

3.16 Mot clé RELA_EFFO_VITE

◊ `RELA_EFFO_VITE`
Mot clé facteur permettant de définir une relation force-vitesse sur un degré de liberté d’un nœud donné sous la forme d’une fonction non linéaire.

Les opérandes `NOEUD`, `SOUS_STRUC`, `NOM_CMP` et `RELATION` ont le même sens pour les mots clés `RELA_EFFO_DEPL` et `RELA_EFFO_VITE`. Ils ne sont donc pas détaillés dans ce paragraphe.

3.17 Mot clé COUPLAGE_EDYOS

◊ `COUPLAGE_EDYOS`
Mot clé facteur obligatoire lors du couplage avec le code EDYOS pour calculer la réponse de machines tournantes sur paliers non-linéaires. Le mot-clé facteur `PALIER_EDYOS` doit alors aussi être renseigné (*cf.* le paragraphe suivant).

3.17.1 Opérande VITE_ROTA

♦ `VITE_ROTA = vrota`
Définition de la vitesse de rotation du rotor en tours par minute.

3.17.2 Opérande PAS_TPS_EDYOS

♦ `PAS_TPS_EDYOS = dtedyos`
Pas de temps initial pour le code EDYOS.

3.18 Mot clé PALIER_EDYOS

◊ `PALIER_EDYOS`
Deuxième mot clé facteur obligatoire lors du couplage avec le code EDYOS. Il permet de définir la position et le type des paliers non-linéaires dont les réponses seront calculées par EDYOS.

Les données à préciser, pour chaque palier non-linéaire, sont :

- position du palier (soit par le nom du nœud du maillage, soit par le nom du groupe contenant ce nœud seul),
- type de palier choisi (correspondant donc à un modèle EDYOS particulier).

Ces données peuvent être entrées suivant deux modes de fonctionnement.
Soit toutes les données sont dans un fichier texte externe, dont l'unité logique sera précisée par le mot-clé UNITE.
Soit on rentre les données dans le fichier de commande directement, *via* GROUP_NO ou NOEUD et TYPE_EDYOS.

3.18.1 Opérande **UNITE**

♦ / UNITE = uled

Définition de l'unité logique du fichier texte externe qui contient la définition des positions et des types de tous les paliers non-linéaires. Voici un exemple de ce fichier :

```
2
1  NOE0   PACONL
2  NOE10   PACONL
```

La première ligne indique le nombre total de paliers non-linéaires, ici deux.
Ensuite, chaque ligne correspond à un palier, avec trois paramètres : numéro (arbitraire) de palier, position du palier (nom du nœud ou du groupe de nœud correspondant) et modèle de palier EDYOS (ici PACONL). Pour le deuxième argument, le code va chercher dans tous les groupes de nœuds du maillage, puis si le nom indiqué n'est pas trouvé, la recherche se poursuit dans les noms de nœuds et en cas d'échec, un message d'erreur est émis et précise que le nom indiqué par l'utilisateur n'a pas pu être trouvé. Le code s'arrête alors en erreur fatale.

3.18.2 Opérande **GROUP_NO**

♦ / GROUP_NO = grnoed

Groupe de nœud ne contenant qu'un seul nœud et qui correspond à la position du palier non-linéaire considéré.

3.18.3 Opérande **NOEUD**

♦ / NOEUD = noed

Nom du nœud qui correspond à la position du palier non-linéaire considéré. On conseille classiquement de privilégier l'utilisation de GROUP_NO à NOEUD pour des raisons de nommage plus pratique.

3.18.4 Opérande **TYPE_EDYOS**

```
♦ / TYPE_EDYOS = /  'PAPANL',
                 /  'PAFINL',
                 /  'PACONL',
                 /  'PAHYNL',
```

Nom du modèle de palier associé, selon la terminologie du code EDYOS. Seuls les types suivants de paliers non-linéaires sont autorisés :

- PAPANL : paliers à patins mobiles,
- PAFINL : palier à patins fixes,
- PACONL : palier avec prise en compte de possibilités de contact,
- PAHYNL : palier hybrides (hydrodynamiques / hydrostatiques avec poches de soulèvement).

3.19 Mot clé **ARCHIVAGE**

◊ ARCHIVAGE

Mot clé facteur définissant l'archivage.

Remarque :

Pour les schémas d'intégration de type Runge-Kutta, l'archivage se fait de manière systématique sur tous les instants de calcul. Ainsi, les valeurs données dans les opérandes `LIST_ARCH`, `INST` et `PAS_ARCH` ne sont pas prises en compte.

3.19.1 Opérande LIST_ARCH/INST

◊ / `LIST_INST = l_arch`

Liste d'entiers définissant les instants de calcul pour lesquels la solution doit être archivée dans le concept résultat `tran_gene`.

◊ / `INST`

Instants de calcul pour lesquels la solution doit être archivée dans le concept résultat `tran_gene` .

3.19.2 Opérande PAS_ARCH

◊ `PAS_ARCH = ipa`

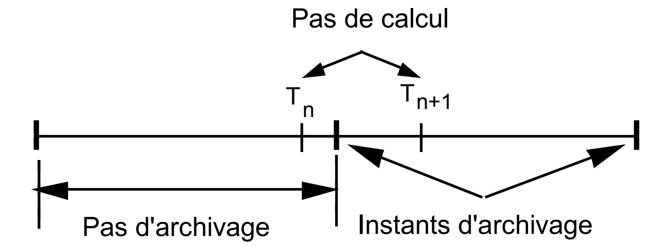
- Méthodes `'EULER'`, `'DEVOGE'`, `'NEWMARK'`, `'ITMI'` :

  Entier définissant la périodicité d'archivage de la solution du calcul transitoire dans le concept résultat `tran_gene`.
  Si `ipa = 5` on archive tous les 5 pas de calcul.
  Quelle que soit l'option d'archivage choisie, on archive le dernier pas de temps et tous les champs associés pour permettre une éventuelle reprise.
  Par défaut on archive tous les pas de calcul.

- Méthodes `'ADAPT_ORDRE1'` et `'ADAPT_ORDRE2'` :

  Entier qui permet de calculer l'intervalle entre deux instants d'archivage dans le concept `résultat`, égal à `PAS_ARCH*PAS`. Avec cette convention, le pas d'archivage est toujours supérieur ou égal au pas maximal utilisé par le calcul.

  Avec un pas variable, les instants d'archivage ne correspondent pas exactement à des pas de calcul. L'algorithme archive donc les grandeurs aux pas de calcul les plus proches des instants d'archivage indiqués par l'utilisateur (en Tn sur ce schéma) :



3.19.3 Opérande CRITERE

◊ `CRITERE =`

Indique avec quelle précision la recherche de l'instant à archiver doit se faire :
`'RELATIF'` : intervalle de recherche `[(1-prec).instant, (1+prec).instant]`
`'ABSOLU'` : intervalle de recherche `[instant-prec, instant+prec]`
La valeur par défaut du critère de recherche est `'RELATIF'`.

3.19.4 Opérande PRECISION

```
◊  PRECISION = /  1.E-06    [DEFAUT]
               /  prec      [R]
```

Indique avec quelle précision la recherche de l'instant à archiver doit se faire.

3.20 Opérande INFO

◊ `INFO = imp`

Entier permettant de préciser le niveau d'impression dans le fichier `MESSAGE`.

Si `INFO=1`, on imprime les informations suivantes dans le fichier `MESSAGE` :

`<I> <nom de la routine où sont écrites les informations suivantes>`

Si `<I> <MDTR74>`, on rappelle que c'est un calcul transitoire sur base modale "classique", sinon `<I> <SSDT74>` c'est un calcul transitoire sur base modale par sous-structuration dynamique.

```
<----------------------------------------------->
Calcul par superposition modale
----------------------------------------------
! la base de projection est un >type de la base de projection<
! le nb d'équations est          : nb
! la méthode utilisée est        : >nom de la méthode d'intégration <
! la base utilisée est           : >nom de la base modale <
! le nb de vecteurs de base est  : nbb

! le pas de temps initial est    : valeur du pas de temps initial
! le nb de pas d'archive est     : nba
! le nombre de lieu(x) de choc est : nbchoc
          (début des informations spécifiques aux schémas adaptatifs)
! nombre de points pas période               :  n bp
! coefficient de remontée du pas de temps    :  cmp
! coefficient de division du pas de temps    :  cdp
! pas de temps minimal                       :  dtmini
! pas de temps maximal                       :  dtmaxi
! nombre maximal de réductions du pas        :  nbred
! vitesse minimale variable      :  >méthode de calcul de la vitesse de référence<
                      (fin des informations spécifiques aux schémas adaptatifs)
! nombre de pas de calcul est     : nbc
! Le pas de temps du calcul est  :  valeur du pas de temps de calcul

! le nombre de RELA_EFFO_DEPL est  : nbrelaed
                      (uniquement si le nombre de relations est non nul)
! le nombre de RELA_EFFO_VITE est  : nbrelaev
                      (uniquement si le nombre de relations est non nul)
----------------------------------------------
```

Si `INFO` : 2, on imprime, en plus des informations écrites dans le cas où `INFO` vaut 1, les informations suivantes dans le fichier `MESSAGE` :

Pour chaque obstacle :

- Le numéro et type de l'obstacle ;
- Le nom et les coordonnées dans le repère global du nœud de choc (des nœuds de choc dans le cas d'un choc entre structures mobiles) ;
- L'orientation, dans le repère global, de la normale à l'obstacle ;
- La valeur de l'angle de vrille ;
- La valeur du jeu initial ;

Et pour chaque nœud de choc et pour chaque mode, le numéro du mode, les valeurs des raideurs locales de choc et du taux de flexibilité locale et de la souplesse locale.
On imprime également à la fin, pour chaque nœud de choc :

`TAUX DE RESTIT FLEXIBILITE : 9.9539E-01` soit 99.53% de souplesse locale ;
`TAUX DE RESTIT EFFORT TRANCHANT : 1.8979E-02` soit 1.89% de l'effort tranchant.

On imprime ces quantité globalement pour l'ensemble des modes et pour chaque mode.

On imprime en outre :

- pour chaque nœud de choc, les rapports souplesse locale sur souplesse de choc et souplesse statique moins souplesse locale sur souplesse de choc,
- pour chaque mode, sa participation sur les déformées statique aux nœuds de choc. Elle vaut le rapport du nombre de conditionnement de la matrice fermée par le vecteur modal et les déformées statiques sur le nombre de conditionnement de la matrice des déformées statiques.

3.21 Opérande IMPRESSION

◊ IMPRESSION
Mot clé facteur qui permet d'imprimer dans le fichier RESULTAT des grandeurs, non imprimables par un opérateur d'impression, telles que le déplacement local, la vitesse locale, les forces de contact aux nœuds de choc et la valeur cumulée sur tous les modes de la base modale de projection du taux de reconstitution de la solution statique.

3.21.1 Opérandes TOUT / NIVEAU

Le mot clé NIVEAU permet d'imprimer un ou plusieurs tableau(x) parmi 'DEPL_LOC', 'VITE_LOC', 'FORC_LOC' et 'TAUX_CHOC'. Avec TOUT = 'OUI' (valeur par défaut), on imprime les quatre tableaux.

3.21.2 Opérandes INST_INIT / INST_FIN

Ces deux mots clés permettent à l'utilisateur de filtrer les impressions dans chaque boucle sur les pas de temps.

3.21.3 Opérande UNITE_DIS_VISC

◊ UNITE_DIS_VISC = unit
Les résultats concernant l'effort, les déplacements visqueux et relatifs entre les deux nœuds, ainsi que la dissipation du dispositif non-linéaire peuvent être sauvegardé dans un fichier directement exploitable par les commandes de *Code_Aster* .

3.22 Opérande TITRE

◊ TITRE = titre
Titre de la structure de données résultat [U4.03.01].

4 Phase d'exécution

4.1 Vérification sur les matrices

Dans le cas d'un calcul par recombinaison modale, on vérifie que les matrices généralisées sont bien issues d'une projection sur une base commune et avec le même nombre de vecteurs de base. Dans le cas d'un calcul par sous-structuration dynamique, on vérifie que les matrices généralisées sont bien issues de la même numérotation généralisée.

4.2 Vérification et conseil sur le choix du pas de temps pour les schémas EULER, DEVOGE et NEWMARK :

On s'assure que le pas de temps choisi vérifie les conditions de stabilité du schéma numérique (critère de CFL) :

- dans le cas de NEWMARK, la stabilité est toujours assurée mais le dépassement du critère peut induire un manque de précision sur le résultat et est signalé par un message ; le calcul se poursuit (au risque de produire un résultat peu précis ou faux).
- dans le cas des schémas d'EULER et DEVOGE, si l'opérande VERI_PAS vaut 'OUI' (valeur par défaut), l'exécution est arrêtée, un pas de temps minimum est proposé. Si l'opérande VERI_PAS vaut 'NON' ou s'il s'agit d'un schéma adaptatif, un message d'alarme est émis et le calcul se poursuit (au risque de produire un résultat peu précis ou faux).

Dans une analyse transitoire sans non-linéarité, il faut veiller à ce que le pas de temps soit tel que :

$dt < 0{,}1/f_n$ pour NEWMARK et DEVOGE

$dt < 0{,}05/f_n$ pour EULER

f_n étant la fréquence la plus élevée des modes de la base modale considérée.

Remarque :

On mentionne qu'avec des non linéarités localisées le pas de temps choisi doit être parfois très inférieur à cette valeur conseillée.

4.3 Phase d'exécution pour les méthodes 'ADAPT_ORDRE1' et 'ADAPT_ORDRE2' :

L'exécution est interrompue lorsque le pas de temps atteint un pas minimal égal à PAS X PAS_LIMI_RELA.

Remarques :

Le schéma des différences centrées ne restitue pas de façon exacte les pulsations propres d'un système, ce qui conduit à d'importantes erreurs de calcul dans les deux cas suivants :

- *Calcul d'un très grand nombre de périodes d'oscillations libres ;*
- *Calcul des oscillations d'un système très faiblement amorti ($\xi < 10^{-3}$) excité sur une fréquence de résonance.*

Dans ces deux cas, il est souvent nécessaire d'augmenter le paramètre NB_POIN_PERIODE .

Les méthodes 'ADAPT_ORDRE1' et 'ADAPT_ORDRE2' peuvent être utilisées en sous-structuration.

Le pas de temps peut être récupéré par l'opérateur RECU_FONCTION , avec la syntaxe suivante :

```
pas = RECU_FONCTION (
        RESU_GENE = dynamoda
        NOM_CHAM = 'PTEM'
        ....)
```

4.4 Phase d'exécution pour la méthode `'ITMI'`

L'exécution est interrompue :

- lorsque la durée d'excitation choisie par l'utilisateur est incompatible avec le temps de simulation souhaité (régime établi + simulation après obtention du régime établi). Dans ce cas, l'utilisateur en est informé avec précision du temps minimum d'excitation requis pour son calcul,
- lorsque l'algorithme ne réussit pas à trouver une solution convergée lors de la diagonalisation de la matrice de raideur,
- lorsque les phases de transition vol/choc ne peuvent être déterminées avec une précision suffisante.